2024 10th International Conference on Control, Decision and Information Technologies
CoDIT 2024 | Valletta, Malta / July 01-04, 2024 Technically co-sponsored by IEEE & IFAC

2024 10th International Conference on Control, Decision and Information Technologies (CoDIT) | 979-8-3503-7397-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/CoDIT62066.2024.10708264

Simulation of Nonlinear Systems Trajectories: between Models and Behaviors*

Antonio Fazzi[1] and Alessandro Chiuso[1]

Abstract— **In this paper, we study connections between the classical model-based approach to nonlinear system theory, where systems are represented by equations, and the nonlinear behavioral approach, where systems are defined as sets of trajectories. In particular, we focus on equivalent representations of the systems in the two frameworks for the problem of simulating a future nonlinear system trajectory starting from a given set of noisy data. The goal also includes extending some existing results from the deterministic to the stochastic setting.**

Keywords: **Nonlinear system theory - Systems representation - Simulation of trajectories**

I. INTRODUCTION

The classical way to solve a problem in the field of system theory is to use a (possibly estimated) system model, that is a set of equations describing the dynamic, to implement the sought design. The difficulties with this approach started whenever the to be modeled systems became complex and / or interconnected, hence the model estimation step became the most computationally expensive part of the whole algorithmic procedure [1]. To deal with this issue, motivated by the increasing quantity of data, the popularity of *data-driven* algorithms increased; such methods design the solution method by learning the system dynamic directly from a set of data (without using explicitly the system equations).

A nowadays popular data-driven approach is the *behavioral setting* [2], where dynamical systems are defined as sets of trajectories (the observed data). Modern applications motivated the extension of such a theory from linear time-invariant (LTI) to classes of nonlinear systems [3], [4], [5], [6]. Such extensions are realized by a LTI embedding of the nonlinear dynamic over a suitable finite dimensional space.

After a brief introduction and a recap of related results, we focus on the problem of simulating trajectories of some nonlinear systems, by looking at the problem from both the data-driven viewpoint and the model-based one. The main contribution to this problem is twofold: propose an iterative data-driven prediction algorithm whose solution is the same as the model-based approach, and extend some existing results from the deterministic to the stochastic setting.

The problem of predicting future system trajectories is the cornerstone of predictive control, thus our analysis provides the basis for nonlinear predictive control, whose solution is currently based on linearization (e.g. [7]).

*This work was not supported by any organization

[1]the authors are with the Department of Information Engineering, University of Padova, via Gradenigo 6/b, I-35131 Padova, Italy antonio.fazzi@unipd.it, alessandro.chiuso@unipd.it

II. NOTATION AND PRELIMINARIES

The definition of a (discrete-time and real valued) dynamical system is a triple $(\mathbb{N}, \mathbb{R}^q, \mathcal{B})$, where $\mathbb{N}$ is the time axis, $\mathbb{R}^q$ is the signal space (q is the number of variables) and $\mathcal{B} \subseteq (\mathbb{R}^q)^{\mathbb{N}}$ is a subset of admissible functions $w : \mathbb{N} \to \mathbb{R}^q$, the *behavior*. The word *admissible* means that the trajectories $w \in \mathcal{B}$ satisfy a difference equation.

A leading role in the behavioral system theory is played by Hankel matrices built from observed trajectories since such matrices are able to *learn* (under suitable assumption, see Lemma 2 below) the system dynamic.

Definition 1: Given a time series $w(t) \in \mathbb{R}^q$, the Hankel matrix with L rows, $H_L(w) \in \mathbb{R}^{qL \times (T-L+1)}$, is

$$H_L(w) = \begin{pmatrix} w(1) & w(2) & \cdots & w(T-L+1) \\ w(2) & w(3) & \cdots & w(T-L+2) \\ \vdots & \vdots & & \vdots \\ w(L) & w(L+1) & \cdots & w(T) \end{pmatrix} \quad (1)$$

.

Lemma 2 ([8]): We are given a LTI system $\mathcal{B}$ with q variables, m inputs and order n, and one of its length-T trajectories $w \in \mathcal{B}$. It holds that rank $H_L(w) = mL + n$ if and only if $\mathcal{B} =$ image $H_L(w)$.

Lemma 2 is a powerful result since it allows to write down all the system trajectories from an observed one. However, the LTI assumption on the system limits its applications. On the same line of [5], we consider single-input single-output (SISO) nonlinear systems of the form:

$$\begin{aligned} \mathcal{B}_{nl} &= \{w = \begin{bmatrix} u \\ y \end{bmatrix} \in (\mathbb{R}^2)^N | \text{ (2) holds}\}, \\ y(t) &= f(x_y(t), x_u(t), t), \quad t = 1, \dots, N \\ x_y(t) &= (\sigma^{-\ell} y(t), \dots, \sigma^{-1} y(t))^T, \\ x_u(t) &= (\sigma^{-\ell} u(t), \dots, \sigma^{-1} u(t), u(t))^T, \\ f(x_y, x_u) &= \theta_{lin} \begin{bmatrix} x_y \\ x_u \end{bmatrix} + \theta_{nl} \phi_{nl}(x_y, x_u). \end{aligned} \quad (2)$$

In (2) u, y are the input (the free variable) and the output (the dependent variable), respectively, ℓ is the system lag (the maximum time shift in the system equation) and N is the trajectory length. $\theta_{lin}, \theta_{nl}$ are parameter vectors, while the finite set of (arbitrary) nonlinear functions $\phi_{nl} = \{\phi_{nl}^1, \dots, \phi_{nl}^K\}$ determines the system dynamic. σ^i is the $i-$th shift operator (the sign of the superscript determines if the shift is backward or forward). For convenience, in the paper, we omit the time dependence.

Remark 3: Systems of the form (2) include the class of SISO output-generalized bilinear systems [6].

The extension of Definition 1 and Lemma 2 to systems of the form (2) reads:

Definition 4: Given a time series $w(t) \in \mathbb{R}^2$ and a finite set of nonlinear functions $\phi_{nl} = \{\phi_{nl}^1, \dots, \phi_{nl}^K\}$, the extended Hankel matrix, $H_{L,\phi_{nl}}(w)$, is

$$H_{L,\phi_{nl}}(w) = \begin{bmatrix} H_L(w) \\ H_L(\phi_{nl}^1(w)) \\ \vdots \\ H_L(\phi_{nl}^K(w)) \end{bmatrix}. \tag{3}$$

By exploiting the LTI embedding of the nonlinear dynamic proposed in [5], we can state the following

Lemma 5: Let $w(t)$ be a T-long trajectory of a system $\mathcal{B}_{nl}$ of the form (2), and assume that $H_{L,\phi_{nl}}(w)$ has rank $(1+K)L+\ell$ with $L \geq \ell+1$. Then, all the trajectories of $\mathcal{B}_{nl}$ are in the range of $H_{L,\phi_{nl}}(w)$.

The rank constraints in both Lemma 2 and 5 hold as far as the data are noise-free. If we work with noise-corrupted data, the involved Hankel matrices appear full rank. In this paper we consider white Gaussian noise on the problem data; the noise corrupted data are generated as

$$\tilde{y}(t) = f(x_{\tilde{y}}(t), x_u(t)) + \mu r(t), \tag{4}$$

where μ is the noise level and $r(t)$ is a random scalar generated from i.i.d. standard normal distributions.

We are going to simulate future trajectories of systems of the form (2) from a set of noisy data. We state the problem first, recalling some properties, and we propose an algorithm then to estimate the solution.

III. Simulation of system trajectories

In this section, we first state the problem of simulating system trajectories from a set of noise-free data and present existing solution methods; we then propose an algorithm to predict the same trajectories from a set of noisy data.

A. Data-driven and model-based simulation

In the behavioral setting, this problem has been studied first in [9] for the class of LTI systems. The solution comes from Willems' fundamental lemma [10]. This milestone result states that it is possible to generate all the finite length system trajectories from an observed longer one. The solution method was recently extended to a class of nonlinear systems in [5] by a finite-dimensional LTI embedding and a consequent modification of the Hankel matrix in order to model the nonlinearities in the data.

The simulation problem we consider follows:

Problem 6: Given a finite-length data trajectory $w_d = col(u_d, y_d) \in \mathbb{R}^2$ of an unknown nonlinear system $\mathcal{B}_{nl}$ of the form (2), a finite set of nonlinear functions ϕ_{nl} (that generates the system equation), an initial condition $w_{ini} = col(u_{ini}, y_{ini}) \in (\mathbb{R}^2)^\ell$ and an input signal $u_f \in \mathbb{R}^{t_f}$, find the output y_f such that the trajectory $\begin{bmatrix} u_f \\ y_f \end{bmatrix} \in \mathcal{B}_{nl}$ and it matches the initial condition.

To deal with Problem 6, [5] restricts to the class of generalized bilinear systems, that are systems whose functions ϕ_{nl} have the form $\phi_{nl}(x_y, x_u) = \psi_{nl}(x_u) + \xi_{nl}(x_u) \otimes x_y$, for some nonlinear functions ψ_{nl}, ξ_{nl}. The solution method relies on embedding the system $\mathcal{B}_{nl}$ into a linear one by adding sets of data transformed via the functions in ϕ_{nl} and exploiting a matrix representation of such special system class, where we can write the added nonlinear inputs as $u_{nl} = \psi + \begin{bmatrix} \Xi_p & \Xi_f \end{bmatrix} \begin{bmatrix} y_{ini} \\ y_f \end{bmatrix}$ (see [5, Lemma 6] for details). The restriction to such a system class involves the ability to simulate the whole trajectory by the solution of one linear system of equations. The solution proposed in [5] follows:

- collect all the nonlinear data terms in a new input variable $u_{nl} = \phi_{nl}(x_y, x_u)$;
- build the Hankel matrices

$$\begin{aligned} H_{\ell+t_f}(u_d) &= \begin{bmatrix} H_\ell(u_d) \\ H_{t_f}(\sigma^\ell u_d) \end{bmatrix}, \\ H_{\ell+t_f}(y_d) &= \begin{bmatrix} H_\ell(y_d) \\ H_{t_f}(\sigma^\ell y_d) \end{bmatrix}, \\ H_{t_f}(u_{d,nl}) &= H_{t_f}(\phi_{nl}(x_{y_d}, x_{u_d})). \end{aligned} \tag{5}$$

- Compute the minimum norm g such that

$$\begin{bmatrix} H_\ell(u_d) \\ H_\ell(y_d) \\ H_{t_f}(\sigma^\ell u_d) \\ H_{t_f}(u_{d,nl}) - \Xi_f H_{t_f}(\sigma^\ell y_d) \end{bmatrix} g = \begin{bmatrix} u_{ini} \\ y_{ini} \\ u_f \\ \psi + \Xi_p y_{ini} \end{bmatrix}. \tag{6}$$

- Result:

$$y_f = H_{t_f}(\sigma^\ell y_d) g. \tag{7}$$

The solution in (6) has been extended to systems that allow nonlinear functions of the past outputs in [6], that is $\phi_{nl}(x_y, x_u) = \psi_{nl}(x_y) + \xi_{nl}(x_y) \otimes x_u$.

Beside the modern data-driven approach for the simulation of system trajectories, a classical way to solve Problem 6 is based on the estimation of a system model. This involves the computation of a set of parameters that define the linear combination among the nonlinear system variables. The solution method requires to build the matrix

$$\begin{bmatrix} x_u(1) & \cdots & x_u(T) \\ x_y(1) & \cdots & x_y(T) \\ \phi_{nl}(x_y(1), x_u(1)) & \cdots & \phi_{nl}(x_y(T), x_u(T)) \\ y(1) & \cdots & y(T) \end{bmatrix}, \tag{8}$$

and then to follow these two steps:

1) Compute the parameters $\hat{\theta}_{lin}, \hat{\theta}_{nl}$ as the left kernel of the matrix in (8) (the solution is unique if the left kernel has dimension one);
2) plug the estimated parameters in (2) to recover the system equation and to compute y.

B. Step-by-step simulation

A natural question is how to deal with Problem 6 whenever the data are corrupted by noise, that is how to simulate trajectories of a general system $\mathcal{B}_{nl}$ of the form (2) from noisy data. We propose an iterative data-driven prediction algorithm that turns out to compute the same estimate as the model-based strategy. The algorithm exploits the link between the initial conditions and the first point of the simulated trajectory. Finite horizon trajectories are estimated by iterating length-one simulations and updating the initial conditions at each step. The problem data are preprocessed first to fulfill the rank constraint in Lemma 5.

We use the following result based on the truncated LQ decomposition.

Lemma 7: Any real matrix A may be decomposed as:

$$A = LQ,$$

where L is a lower triangular matrix and Q is a unitary matrix[1]. By writing the block factorization

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \approxeq \begin{bmatrix} L_{11}Q_1 \\ L_{21}Q_1 \end{bmatrix} = A^*, \quad (9)$$

we get that the matrix A^* is a low-rank approximation of A. The rank reduction is the row dimension of Q_2.

The proposed algorithm for the prediction of future trajectory from noisy data of the form (4) is shown in Algorithm 1. It uses Lemma 7 to estimate the Hankel data matrix and iterates (6) to predict the points of the trajectory.

The difference between Algorithm 1 and the approaches in [5], [6] is that an iterative strategy imposes no restriction on the system class. We are going to show how Algorithm 1 strongly connects the model-based approach to the behavioral theory (at least for the considered prediction problem).

Theorem 8: The iterative data-driven prediction of Algorithm 1 estimates the same solution as the model-based approach.

Proof: We show that one step of Algorithm 1 and one step of the model-based prediction (briefly recalled in the deterministic case in Section III-A), starting from the same data, initial conditions and *future* input signal, estimate the same point. The result holds true by assuming that we use the same set of basis functions ϕ_{nl} and that the data matrix is approximated by the LQ decomposition (see Lemma 7).

Model-based: Following (8), we need to estimate the system parameters $\theta_{lin}, \theta_{nl}$ as a (approximate) left null space of the matrix

$$\underbrace{\begin{bmatrix} x_u(1) & \cdots & x_u(T) \\ x_{\tilde{y}}(1) & \cdots & x_{\tilde{y}}(T) \\ \phi_{nl}(x_{\tilde{y}}(1), x_u(1)) & \cdots & \phi_{nl}(x_{\tilde{y}}(T), x_u(T)) \\ \tilde{y}(1) & \cdots & \tilde{y}(T) \end{bmatrix}}_{\mathcal{S}}. \quad (12)$$

Since the outputs are noisy, the matrix $\mathcal{S}$ is full rank. We impose the existence of a left kernel by projecting the last row on the space generated by the data $x_u, x_{\tilde{y}}$ and their nonlinear transformations. This is done using Lemma 7:

$$\mathcal{S} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}. \quad (13)$$

The model-based estimator is then given by the projection of the row $\tilde{y}$ onto the space generated by the (noisy) rows $x_u, x_{\tilde{y}}, \phi_{nl}(x_{\tilde{y}}, x_u)$ by neglecting the term $L_{22}Q_2$ in (13):

$$\hat{y}_{MB} = \mathcal{P}_{x_u, x_{\tilde{y}}, \phi_{nl}(x_{\tilde{y}}, x_u)}\ (\tilde{y}) = L_{21}Q_1 = L_{21}L_{11}^{-1}L_{11}Q_1.$$

If we set $[\hat{\theta}_{lin}, \hat{\theta}_{nl}] = L_{21}L_{11}^{-1}$, the one-step model-based estimator can be written as

$$\hat{y}_{MB} = [\hat{\theta}_{lin}, \hat{\theta}_{nl}] v_{ini,i}, \quad \text{(MB)}$$

where $v_{ini,i}$ is the vector in the right-hand side of (11).

Data-driven (Algorithm 1-one step): the key observation is that the matrix $\mathcal{S}$ in (8) and the block Hankel matrix $\begin{bmatrix} H_d \\ H_1(\sigma^\ell \tilde{y}_d) \end{bmatrix}$ are the same, up to a permutation of the rows (this happens since $t_f = 1$). Following Algorithm 1, any length-one data-driven prediction can be computed as follows:

1) Project $H_1(\sigma^\ell \tilde{y}_d)$ onto the rows generated by the data (see line 3 of Algorithm 1) and get $\bar{y} = H_1(\bar{y})$;

[1]the LQ decomposition can be computed from the (more popular) QR decomposition

Algorithm 1 Data-driven prediction for nonlinear systems

```
Require: $u_d, \tilde{y}_d$  $d = 1, \ldots, n_d$ (data points), $w_{ini}$ initial
    condition, $\ell$ (system lag), $T_f$ (prediction horizon), $u_f$
    (future input), $\phi_{nl}$ (set of nonlinear functions)
Ensure: $y_f$ (predicted output)
1: Build the Hankel matrices as in (5) with $t_f = 1$:
    $$H_{\ell+1}(u_d) = \begin{bmatrix} H_\ell(u_d) \\ H_1(\sigma^\ell u_d) \end{bmatrix},$$
    $$H_{\ell+1}(\tilde{y}_d) = \begin{bmatrix} H_\ell(\tilde{y}_d) \\ H_1(\sigma^\ell \tilde{y}_d) \end{bmatrix}$$
    $$H_1(\tilde{u}_{d,nl}) = \begin{bmatrix} H_1(\phi_{nl}(x_{\tilde{y}_d}, x_{u_d})) \end{bmatrix}.$$
2: Build
    $$H_d = \begin{bmatrix} H_\ell(u_d) \\ H_\ell(\tilde{y}_d) \\ H_1(\sigma^\ell u_d) \\ H_1(\phi_{nl}(x_{\tilde{y}_d}, x_{u_d})) \end{bmatrix} \quad (10)$$
3: Use Lemma 7 to project $H_1(\sigma^\ell \tilde{y}_d)$ onto the rows of the
    data matrix $H_d$ (10)
    $$H_1(\bar{y}_d) = \mathcal{P}_{H_d}(H_1(\sigma^\ell \tilde{y}_d))$$
4: for $i = 1 : T_f$ do
5:     Compute the vector $g_i$ of minimum norm that ap-
       proximately satisfies the system
       $$H_d g_i \approx \underbrace{\begin{bmatrix} u_{ini,i} \\ y_{ini,i} \\ u_{f,i} \\ \phi_{nl}(y_{ini,i}, \begin{bmatrix} u_{ini,i} \\ u_{f,i} \end{bmatrix}) \end{bmatrix}}_{v_{ini,i}}, \quad (11)$$
       where the subscript $i$ denotes the iteration step
6:     Set $\hat{y}_i = H_1(\bar{y}_d) g_i$
7:     Store the $i$-th pair $(u_i, \hat{y}_i)$
8:     Set $(u_i, \hat{y}_i)$ as the last initial condition
9: end for
```

2) compute g_i that solves the least squares problem in (11);
3) set $\hat{y}_{DD} = \bar{y} g_i$.

Since the vector g_i defines a linear combination of the columns of the data matrix, because of the projection in the first step, there exists γ such that $g_i = Q_1^T \gamma$ (where Q_1 is the matrix in (13)) [11]. Doing so, we get

$$v_{ini,i} \approx H_d Q_1^T \gamma = L_{11} Q_1 Q_1^T \gamma = L_{11} \gamma. \tag{14}$$

The second block equation in (13) gives

$$\hat{y}_{DD} = \bar{y} g_i = (L_{21} Q_1 + L_{22} Q_2) g_i = (L_{21} Q_1) Q_1^T \gamma = L_{21} \gamma, \tag{15}$$

because of the orthogonality between Q_1 and Q_2 (that follows from the projection step obtained with the approximate LQ decomposition). The thesis holds true since $\gamma = L_{11}^{-1} v_{ini,i}$ (see (14)). ■

A series of remarks about Algorithm 1 follows:

- **Deterministic setting** In the deterministic setting, the solution computed by Algorithm 1 matches the correct output for the systems of the form (2). In particular, this holds for the output generalized bilinear systems presented in [6];
- **Model-based vs data-driven** Theorem 8 shows an equivalence between the model-based and the behavioral approach for the prediction of future trajectories. This comes from the computational choices in Algorithm 1;
- **Computational cost** The computations in Algorithm 1 require the solution of T_f least squares problems of the form (11). The computational cost can be reduced by observing that all the linear systems have the same data matrix and only the right hand side (initial conditions and input) changes at each iteration [12].

A natural question arises: is it still worth adopting an iterative computational strategy whenever it is possible to simulate the trajectory all at once (that is, *e.g.*, for the class of output-generalized bilinear systems in [6])? Our goal is not to answer this question, but we simply proposed a way to predict the trajectories of an estimated system, by shortening the distance between two apparently different approaches.

IV. Connection between behaviors and system models

The common characterization of data-driven algorithms is that they do not need to write down explicitly a system model (a set of equations) to develop a solution method; a set of data is enough to learn the system dynamics. This is a feature of the behavioral theory too, where systems are represented (in a data-driven fashion) by rank constrained block Hankel matrices built from observed trajectories. However, this is simply a different way of writing down a system, since the rank constraint on the Hankel matrix is equivalent to the existence of a system representation (an equation whose coefficients are given by the left kernel of the matrix), that is *hidden* in the left kernel of such a matrix. Thus, there are two different but interconnected ways of approaching the same problem. Both the approaches can be valid alternatives, and studying their properties and connections (mainly in the field of control problems) is a research topic by itself [13], [14], [15], [16].

In the behavioral theory (despite being a data-driven approach), system representations exist and they can be useful. The rank constraints on the Hankel matrices enforce the existence of a system representation (commonly known as kernel representation), and vice versa. Indeed, the definition of a dynamical system $\mathcal{B}$, in general, involves a difference equation defined by a polynomial operator R:

$$\mathcal{B} = \left\{ w \in \mathbb{R}^q | R(\sigma) w(t) = 0 \right\},\ t = 1, \ldots, T, \tag{16}$$

where $\sigma(w(t)) = w(t+1)$ is the shift operator. We read in (16) that the trajectories w of the system $\mathcal{B}$ and its (non-unique) representation R, given by the finite-length snapshots of $w(t)$ and the rows of the operator R, respectively, are two *distinct* but *interconnected* objects. They represent the data-based and the model-based approach to the same problem, respectively, and they are linked by (16).

We can look at some examples from the literature, where the same problem is solved by these two different approaches (restricted to the case of LTI behaviors):

- the distance between LTI systems: [17] is based on the Hankel matrices generated by the two system trajectories, while [18] or [19] are based on the systems representations only (being representation invariant, though);
- controllability test: [20] is based on Hankel matrices, while [21] relies on a coprimeness test between the polynomials in the system representations.

The row dimension of R plays an important role in the prediction problem, and in particular in the length of the to-be-computed trajectory. R is said *minimal* if its rows are linearly independent [22]. Each linearly dependent row in R increases by one the length of the predicted trajectory. Thus, increasing the row size of the Hankel matrix can be viewed as adding (linearly dependent) equations in a vectorial system model. Observe that, in this last case, the rank of the Hankel matrix does not change (but the dimension of its null space does).

V. Examples

In the following, we are going to run and analyze some examples of estimation of future trajectories computed by Algorithm 1. The analysis will focus on different perturbations on the same set of noise-free data, to observe and discuss how the results change for a given set of nonlinear functions ϕ_{nl}.

The system we consider is given by the difference equation

$$y(t+2) = \sin(y(t+1)) - .1 y(t)^2 + u(t). \tag{17}$$

whose lag is $\ell = 2$. The setup for the experiment follows:

1) build a trajectory of 80 points for the system (17) starting from $(1, 1)$ and a random input signal;
2) length-68 problem data: generate 100 random input signals and the corresponding noisy outputs as in (4) with $\mu = 0.1$;

3) initial conditions: $\begin{pmatrix} u_{69} \\ u_{70} \end{pmatrix}$, $\begin{pmatrix} y_{69} \\ y_{70} \end{pmatrix}$ from point 1 (they are the same in all the 100 experiments);
4) for each of the 100 data pairs $(u, \tilde{y})$ from point 2, use the given set of basis functions, the initial conditions from point 3 and the last ten inputs from point 1 to predict a trajectory (by using Algorithm 1).

Remark 9: The noise-free data (first 68 inputs and outputs) in point 1 are not useful, but we used them only to generate the initial conditions (and an exact, but unknown, trajectory to be compared with the results of the experiment). The goal of the experiment is to observe and analyze the results of the statistics over the 100 solutions computed by Algorithm 1, comparing them with the true system trajectory. To make the example more realistic, we consider several nonlinear functions to (hopefully) include the ones in the system equation (or to well approximate them, at least). In particular, we consider

$$\phi_{nl} = \{y^2, y^3, y^4, \cos y, \sin y, e^y\}.$$

Remark 10: In the deterministic setting, the addition of linearly independent functions to the set ϕ_{nl} does not change the result of the prediction problem. This is because

- in the system equation, the corresponding parameter is set to zero;
- in the block-Hankel matrix representation, while the row dimension of the matrix increases, the rank loss is the same.

For this reason, we did not add the input u as an argument of the nonlinear functions in ϕ_{nl}.

Considering a bigger set of functions ϕ_{nl} than the minimal one in the stochastic setting has two main consequences:

1) the noise spreads through more (nonlinear) terms;
2) bigger block-Hankel matrices usually means ill-conditioned problems.

Estimating the system from a set of functions ϕ_{nl} bigger than the minimal one means *choosing* a particular system among a bigger model class defined by all the considered basis functions. The addition of a regularization term in the solution of the least squares problem (11) enforces the identification of the sought system.

The regularization in system identification is a classical approach in the literature [23], and different choices for the norm of the regularization term already appeared in system identification problems [24], [25], [26]. We consider the following two choices:

$$\min_{g_i} \|H_d g_i - v_{ini,i}\|_2 + \lambda_1 \|g_i\|_1, \tag{18}$$

$$\min_{g_i} \|H_d g_i - v_{ini,i}\|_2 + \lambda_2 \|g_i\|_2, \tag{19}$$

where λ_1 and λ_2 are regularization parameters. In the following experiments, we solve all the 100 problems for different values of λ_1 and λ_2 and we choose the one for which the average over the 100 estimated trajectories is closer (in norm) to the exact one.

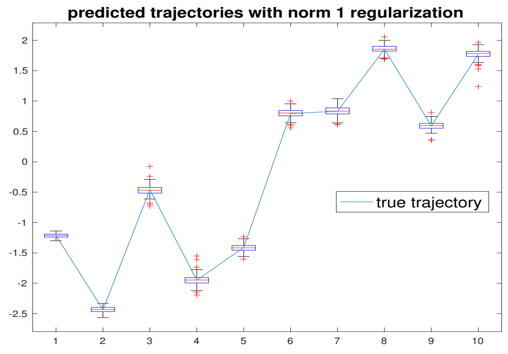

Fig. 1. Estimation of a trajectory of (17) from noisy data and a bigger set of basis functions: norm-1 regularization.

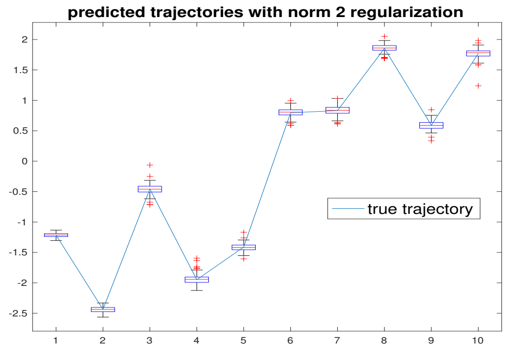

Fig. 2. Estimation of a trajectory of (17) from noisy data and a bigger set of basis functions: norm-2 regularization.

The results for the two choices are shown in Figures 1 and 2, respectively.

Both experiments show that most of the predicted trajectories are very close to the exact (but unknown) one. Indeed, the two figures look similar. The choice between (18) and (19) corresponds to a different weighting of the entries of the solution vector. The advantage of (19) can be the easier implementation.

To key points of the experiments follow:

1) the choice of the basis functions in ϕ_{nl} influences the accuracy of the computed trajectories (that is, how well they approximate the exact solution). The preliminary knowledge of such functions is a strong assumption;
2) the addition of regularization in the solution method can deal with an increasing number of basis functions as well as with ill-conditioned problems;
3) the choice of the norm in the regularization can influence the accuracy of the estimated solution (this did not happen In the considered example because of the small problem size and the low noise level).

VI. Applications in Control

As well as being important itself, the proposed prediction algorithm can be the key tool in the solution of some nonlinear control problems in the stochastic setting. We briefly sketch some ideas in the following.

A. *Output matching problem*

The output matching problem for output-generalized bilinear systems has been studied in [6] in the deterministic setting. Roughly speaking, such a problem consists in the analogous of a prediction problem, where the role of inputs and outputs is switched. So we aim to compute the inputs that generated a given output trajectory.

The extension of this problem to the stochastic setting can be straightforward by using the prediction algorithm proposed in the paper.

B. *Nonlinear predictive control*

The estimation of future system trajectories is the cornerstone of predictive control strategies. This approach consists of iterating the following steps:

1) predict N future steps from the given set of data;
2) optimize the estimated steps with respect to a cost function;
3) get an optimal controller from the optimization process;
4) keep the first input value from the controller and go to the next iteration.

If the system is represented by a (estimated) equation, this problem is commonly known as MPC (Model Predictive Control). But, it is possible to solve the same problem by replacing the system equation with a set of (noisy) data. The data-driven prediction algorithm proposed in the paper can be used for the first step. Optimization strategies for the data-driven setting are currently object of research.

VII. Conclusion

Moving on the line between model-based and data-driven approaches for the theory of nonlinear systems, we proposed an iterative data-driven algorithm that predicts the same solution as the model-based approach from a set of noisy data. The goal was twofold: show some connection between the two (apparently) different strategies and extend some existing results from the deterministic to the stochastic setting.

The proposed prediction algorithm can be useful in the solution of some nonlinear control problems. Future work can focus on the analysis and development of nonlinear control methods (in particular, the one described in Section VI-B).

ACKNOWLEDGMENT

A. Fazzi is member of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM).

References

[1] H. Hjalmarsson, “From experiment design to closed-loop control,” *Automatica*, vol. 41, no. 3, pp. 393–438, 2005.

[2] J. W. Polderman and J. C. Willems, *Introduction to Mathematical Systems Theory*, ser. Texts in Applied Mathematics. New York, NY: Springer New York, 1998, vol. 26.

[3] V. K. Mishra, I. Markovsky, A. Fazzi, and P. Dreesen, “Data driven simulation for NARX systems,” in *2021 29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 1055–1059.

[4] J. G. Rueda-Escobedo and J. Schiffer, “Data-driven internal model control of second-order discrete volterra systems,” in *IEEE Conf. Decision Control*. IEEE, 2020, pp. 4572–4579.

[5] I. Markovsky, “Data-driven simulation of generalized bilinear systems via linear time-invariant embedding,” *IEEE Trans. Automat. Contr.*, 2023.

[6] L. Hemelhof, I. Markovsky, and P. Patrinos, “Data-driven output matching of output-generalized bilinear and linear parameter-varying systems,” in *2023 European Control Conference (ECC)*, 2023.

[7] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, “Linear Tracking MPC for Nonlinear SystemsPart II: The Data-Driven Case,” *IEEE Transactions on Automatic Control*, vol. 67, no. 9, pp. 4406–4421, 2022.

[8] I. Markovsky and F. Dörfler, “Identifiability in the behavioral setting,” *IEEE Trans. Automat. Contr.*, 2023.

[9] I. Markovsky and P. Rapisarda, “Data-driven simulation and control,” *Int. J. Control*, vol. 81, pp. 1946–1959, 2008.

[10] J. C. Willems, P. Rapisarda, I. Markovsky, and B. De Moor, “A note on persistency of excitation,” *Syst. Control Lett.*, vol. 54, no. 4, pp. 325–329, 2005.

[11] V. Breschi, A. Chiuso, and S. Formentin, “Data-driven predictive control in a stochastic setting: a unified framework,” *Automatica*, vol. 152, p. 110961, 2023.

[12] G. Golub and C. Van Loan, *Matrix Computation*, 3rd ed. Baltimore, Maryland: Johns Hopkins University Press, 1996.

[13] A. Chiuso, M. Fabris, V. Breschi, and S. Formentin, “Harnessing the final control error for optimal data-driven predictive control,” 2023, https://arxiv.org/abs/2312.14788.

[14] F. Dörfler, P. Tesi, and C. De Persis, “On the certainty-equivalence approach to direct data-driven lqr design,” *IEEE Trans. Automat. Control*, pp. 1–8, 2023.

[15] V. Krishnan and F. Pasqualetti, “On direct vs indirect data-driven predictive control,” in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 736–741.

[16] F. Dörfler, J. Coulson, and I. Markovsky, “Bridging direct & indirect data-driven control formulations via regularizations and relaxations,” *IEEE Trans. Automat. Control*, vol. 68, pp. 883–897, 2023.

[17] A. Padoan, J. Coulson, H. J. van Waarde, J. Lygeros, and F. Dörfler, “Behavioral uncertainty quantification for data-driven control,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 4726–4731.

[18] A. Fazzi and I. Markovsky, “Distance problems in the behavioral setting,” *Eur. J. Control*, vol. 74, p. 100832, 2023, 2023 European Control Conference Special Issue.

[19] K. De Cock and B. De Moor, “Subspace angles between ARMA models,” *Syst. Control Lett.*, vol. 46, pp. 265–270, 2002.

[20] V. Mishra, I. Markovsky, and B. Grossmann, “Data-driven tests for controllability,” *Control Systems Letters*, vol. 5, p. 517522, 2020.

[21] A. Fazzi, N. Guglielmi, and I. Markovsky, “Computing common factors of matrix polynomials with applications in system and control theory,” in *Proc. of the IEEE Conf. on Decision and Control*, Nice, France, 2019, pp. 7721–7726.

[22] J. C. Willems, “From time series to linear system- Part I. Finite dimensional linear time invariant systems,” *Automatica*, vol. 22, pp. 561–580, 1986.

[23] G. Pillonetto, T. Chen, A. Chiuso, G. De Nicolao, and L. Ljung, *Regularized System Identification: Learning Dynamic Models from Data*, ser. Communications and Control Engineering. Springer International Publishing, 2022.

[24] I. Markovsky, “Application of low-rank approximation for nonlinear system identification,” in *25th IEEE Mediterranean Conference on Control and Automation*, Valletta, Malta, July 2017, pp. 12–16.

[25] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[26] J. Suykens and J. Vandewalle, “Least Squares Support Vector Machine Classifiers,” *Neural Processing Letters*, vol. 9, p. 293300, 1999.